Volume 2, Issue 7, July 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Comparative Analysis & Evaluation of Euclidean Distance Function and Manhattan Distance Function Using K-means Algorithm

Amit Singla
University Institute of Engineering and Technology
Kurukshetra University Kurukshetra
amit3singla@gmail.com

Mr. Karambir
Assistant Professor
UIET,Kurukshetra University,Kurukshretra
bidhankarambir@rediffmail.com

Abstract— **Clustering is division of data into groups of similar objects. Each group, called a cluster, consists of objects which are similar between themselves and different as compared to objects of the other groups. In cluster, analysis is the organization of a collection of patterns into cluster based on similarity. This paper is intended to study and compare Euclidean distance function and Manhattan distance function by using k-means algorithm. This distance functions are compared according to number of iterations and within sum of squared error. Some conclusions that are extracted belong to the time complexity and accuracy.**

Keywords— **k-means algorithm, Euclidean distance function, Manhattan distance function, weka tool, clustering, time complexity.**

I. INTRODUCTION

Clustering is considered the most important unsupervised learning problem; so, as every other problem of this kind, it deals with finding a structure in a collection of unlabeled data. A clustering could be "the process of organizing objects into groups whose members are similar in some way". A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. A cluster of data objects can be treated collectively as one group and so may be considered as a form of data compression. Although classification is an effective means for distinguishing groups or classes of objects, it requires the often costly collection and labeling of a large set of training tuples or patterns, which the classifier uses to model each group. It is often more desirable to proceed in the reverse direction: First partition the set of data into groups based on data similarity (e.g., using clustering), and then assign labels to the relatively small number of groups. Additional advantages of such a clustering-based process are that it is adaptable to changes and helps single out useful features that distinguish different groups.

II. K-MEANS CLUSTERING ALGORITHM

K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume

k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point need to re-calculate k new centroids as bar centers of the clusters resulting from the previous step. After these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.

Finally, this algorithm aims at minimizing an objective function, in this case a squared error function.

The algorithm is composed of the following steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into

groups from which the metric to be minimized can be calculated.

The reason behind choosing k-means algorithm:

- Its time complexity is O(nkl), where n is the number of patterns, k is the number of clusters and l is the number of iteration taken by algorithm to coverge.
- Its space complexity is O(k+n).It requires additional space to store the data matrix.
- It is order independent; for a given initial seed set of cluster centers, it generates the same partition of the data irrespective of the order in which the patterns are presented to the algorithm.

III. DISTANCE MEANSURES

An important component of a clustering algorithm is the distance measure between data points. The problem arises from the mathematical formula that are used to combine the distances between the single components of the data feature vectors into a unique distance measure that can be used for clustering purposes: different formulas leads to different clustering.

The most popular distance measure :

A. Euclidean distance function

In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" distance between two points that one would measure with a ruler, and is given by the Pythagorean formula. By using this formula as distance, Euclidean space becomes a metric space.

B. Manhattan distance function

The Manhattan distance function computes the distance that would be travelled to get from one data point to the other if a grid-like path is followed. The Manhattan distance between two items is the sum of the differences of their corresponding components.

IV. WEKA TOOL

WEKA is the product of the University of Waikato (New Zealand) and was first implemented in its modern form in 1997. It uses the GNU General Public License (GPL). The software is written in the Java™ language and contains a GUI for interacting with data files and producing visual results (think tables and curves). It also has a general API, so you can embed WEKA, like any other library, in our own applications to such things as automated server-side data mining tasks. Weka supports several standard data mining tasks, more specifically, data pre-processing, clustering, classification, regression, visualization, and feature selection.

The reason behind choosing this software:

- This is the most popular software for implementing different clustering algorithms.
- This tool is very powerful in implementing different data clustering algorithms.
- Weka also provides the graphical user interface of the user and provides many facilities.

V. IMPLIMENTATION AND RESULTS

In this paper, for the purpose of data mining Weka 3.7 version software used and for experimentation QuantumChemistry_test_blind.arff is used. Different distance function regarding number of iterations and sum of squared errors were compared by increasing the number of cluster. By increasing number of cluster author observed within sum of squared error get reduced which shows greater the number of cluster smaller the within sum of squared error.

Table1 and Table2 shows the experiment result the number of iteration in Manhattan distance function are greater as compared to Euclidean distance function except k = 64 number of iterations of Manhattan distance function is equal to number of iteration of Euclidean distance function. As in k-means algorithm, the time complexity is directly proportional to the number of iterations and the findings shows that number of iterations of Euclidean distance function is less so it takes computational time as compared to Manhattan distance function but for k=64 computational time is less in case of Manhattan distance function.

TABLE I

EXPERIMENT RESULT OF NUMBER OF ITERATION IN EUCLIDEAN DISTANCE FUNCTION AND MANHATTAN DISTANCE FUNCTION

k	**No. of iteration in Euclidean distance**	**No. of iteration in Manhattan distance**
8	4	8
16	5	6
32	4	4
64	4	3

TABLE 2

EXPERIMENT RESULT OF WITHIN SUM OF SQUARED ERRORS IN EUCLIDEAN DISTANCE FUNCTION AND SUM OF WITHIN CLUSTER DISTANCES IN MANHATTAN DISTANCE FUNCTION

k	**Within cluster sum of squared errors in Euclidean distance**	**Sum of within cluster distances in Manhattan distance**
8	41.10	232.55
16	25.93	180.42
32	17.62	133.69
64	7.58	82.52

VI. CONCLUSIONS

Cluster analysis groups various objects based on their similarity. Clustering analysis is the pivot for data mining. K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. K-means algorithm aims at minimizing an objective function, in this case a squared error function. The purpose of this experiment is to find the effect of distance functions on clustering. Euclidean distance function and Manhattan distance function were used to see this effect. Experimental shows that within sum of squared error get reduced with the increasing value of k. Results also revealed that the number of iteration in Manhattan distance function are greater as compared to Euclidean distance function except k = 64 whereas for k = 32 both distance functions shows same result. As in k-means algorithm, the time complexity is directly proportional to the number of iterations and the findings shows that number of iterations of Euclidean distance function is less so it takes computational time as compared to Manhattan distance function but for k=64 computational time is less in case of Manhattan distance function. Different approaches were used to measure the distance among various data objects which is the most significant step of creating cluster. So special consideration should be given to choose distance function and it should be chosen according to data set and number of cluster.

REFERENCES

[1] Zhang T., Ramakrishan R., "BIRCH: A New Data Clustering Algorithm and Its Applications", pp.103-114, 1997.

[2] Li C. , Biswas G., "Unsupervised learning with mixed numeric and nominal data" vol. 14, No. 4, pp. 676-690, July/August 2002..

[3] Samatov N., OSTROUCHOV G. & GEIST A., "RACHET: An Efficient Cover-Based Merging of Clustering Hierarchies from Distributed Datasets", pp. 157–180 , 2002.

[4] Liao T., "Clustering of time series data-a survey", pp. 1857-1874, January 2005..

[5] Al-Omari F., Al-Fayoumi N., "IMDC: An Image-Mapped Data Clustering Technique for Large Datasets", pp. 112-115, 2005.

[6] Hassan R., Nath B., Kirley M. , "A Data Clustering Algorithm Based On Single Hidden Markov Model" , pp. 57 – 66 , 2006.

[7] Shih M., Jheng J.,Lai L. introduced the "A Two-Step Method for Clustering Mixed Categroical and Numeric Data", Tamkang Journal of Science and Engineering, Vol. 13, No. 1, pp. 11-19, 2010.

[8] Chauhan R., Kaur H., "Data Clustering Method for Discovering Clusters in Spatial Cancer Databases", vol. 10 , No. 6, November 2010.

[9] Gothai, E., Balasubramanie P. "An Efficient Way for Clustering Using Alternative Decision Tree", American Journal of Applied Sciences, pp. 531-534, 2012.

[10] Wang S., Dutta H. "PARABLE: A Parallel Random-partition Based Hierarchical Clustering Algorithm for the MapReduce Framework" , 2011.

[11] Abbas O., "Comparison between Data Clustering Algorithms" The International Arab Journal of Information Technology , vol. 5, No. 3 , July 2008.

[12] Jain A., Murthy M. & Flynn P., "Data Clustering: A Review" vol. 31, No. 3, September 1999.

